J Sci Comput (2013) 55:372–391
DOI 10.1007/s10915-012-9636-1

Numerical Solution of the Kohn-Sham Equation by Finite Element Methods with an Adaptive Mesh Redistribution Technique

Gang Bao · Guanghui Hu · Di Liu

Received: 20 February 2012 / Revised: 18 August 2012 / Accepted: 20 August 2012 /
Published online: 1 September 2012

Abstract A mesh redistribution method is introduced to solve the Kohn-Sham equation. The standard linear finite element space is employed for the spatial discretization, and the self-consistent field iteration scheme is adopted for the derived nonlinear generalized eigenvalue problem. A mesh redistribution technique is used to optimize the distribution of the mesh grids according to wavefunctions obtained from the self-consistent iterations. After the mesh redistribution, important regions in the domain such as the vicinity of the nucleus, as well as the bonding between the atoms, may be resolved more effectively. Consequently, more accurate numerical results are obtained without increasing the number of mesh grids. Numerical experiments confirm the effectiveness and reliability of our method for a wide range of problems. The accuracy and efficiency of the method are also illustrated through examples.

Keywords Adaptive mesh redistribution · Harmonic map · Finite element method · Kohn-Sham equation · Density functional theory

1 Introduction

The Schrödinger equation is fundamental for the quantum mechanical description of electronic structures of matter, since it does not require any empirical input [32]. The time-independent Schrödinger equation takes the following form

$$H\Psi = E\Psi, \tag{1}$$

G. Bao
Department of Mathematics, Zhejiang University, Hangzhou 310027, China
e-mail: bao@math.msu.edu

G. Bao · G. Hu (✉) · D. Liu
Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA
e-mail: ghhu@math.msu.edu

D. Liu
e-mail: richardl@math.msu.edu

where H is the Hamiltonian, and E and Ψ represent the energy eigenvalue and eigenstate (wavefunction), respectively. Under most circumstances, it is too expensive to numerically solve Eq. (1) because of the high dimensionality of the wavefunction Ψ. Therefore computationally more feasible models for electronic structures become very important for practical simulations.

There are several computable models available for the electronic structure calculation such as the Hartree-Fock method [6], the quantum Mento-Carlo method [27], the Kohn-Sham density functional theory [25], and the orbital-free density functional theory [3, 14]. Among these models, the Kohn-Sham density functional theory has the advantage that the Hohenberg-Kohn theorem [10] theoretically guarantees the ground-state electron density to uniquely determine all properties of an electronic system. To numerically solve the Kohn-Sham equation, there are many methods in the market, such as the planewave expansion method [20], the finite difference method [8], the finite element method [21, 29, 32], the wavelet method [35], the discontinuous Galerkin method [24], and the mesh-free method [33]. In this paper, we focus on the finite element method because of its attractive features. For example, by using the finite element method, boundary conditions of various types can be handled straightforwardly. Also, the physical domain with a complex geometry can also be handled without technical difficulty, which makes the finite element method more suitable for the practical simulation.

In previous works about finite element methods, it is indicated that a radial mesh seems to be a good choice for solving the Kohn-Sham equation because of the existence of the external potential. In the Kohn-Sham Hamiltonian, the external potential describes the attraction between the nucleus and the electron. This term is very singular in the vicinity of the nucleus. A radial mesh has small mesh size in the vicinity of each nucleus, and large one in the region away from nucleus. With this configuration of mesh grids, both the numerical accuracy and the efficiency could benefit because less mesh grids are needed to effectively solve the singular external potential, compared with the uniform mesh case. Two well-designed non-uniform meshes are given in [32] and [21] respectively, and excellent numerical results are presented there. But meshes in [32] and [21] are generated beforehand and remain static during the simulation. In this paper, we are going to adopt a mesh adaptive technique that provides a possible way to dynamically generate a radial mesh. It means that the distribution of the mesh grids and/or the mesh topology could be adjusted during the simulation according to, for example, the profile of the electron density. This is advantageous when the structure of the electronic system or the physical domain is complex and evolves with time. In such case, to generate a quality mesh in advance is not trivial.

There are three classical adaptive methods: the h-adaptive methods which locally refine and coarsen the mesh, the r-adaptive methods which redistribute the grid points while keeping the number of mesh grids unchanged, and the p-adaptive methods which locally enrich the order of the approximate polynomial. The r-adaptive method is also known as the moving mesh method, which has been widely used in a variety of fields such as the computational fluid dynamics [34, 38], phase-field model [2, 40], reaction-diffusion models [12], unstable flow in porous media [11]. The reviews of the moving mesh method can be found in [1, 13]. An interesting discussion is found in [35] for numerically solving the Kohn-Sham equation: "*The question of the feasibility of molecular dynamics with an unstructured nonuniform FE-mesh is often raised. The FE-community provides the following suggestion, based on experience in other fields: each atom is attached to the mesh, and the mesh is considered to be made of 'rubber'. As the atom moves, the mesh moves along…*". Based on our numerical experience, r-adaptive methods can partially satisfy the above requirements. A motivation of our current work is to propose a framework for solving the Kohn-Sham equation by using

the r-adaptive methods. For the early work on this topic, we refer to [37] and references therein, where the curvilinear coordinate method is utilized to improve the mesh quality.

In this paper, we present a finite element solver with an r-adaptive technique to simulate the Kohn-Sham equation with tetrahedral meshes. The solver consists of two main iterations. The first one is a self-consistent field (SCF) iteration to generate the self-consistent electron density on the current mesh, while the second iteration is adopted to adaptively optimize the distribution of mesh grids in terms of the self-consistent electron density. The Kohn-Sham equation is discretized by the standard finite element method. After the self-consistent electron density is obtained, the moving mesh method is implemented to improve the mesh quality. The idea of our moving mesh method was originally proposed by Dvinsky [5] based on the observation that a uniform mesh can be mapped to a nonuniform mesh via the Harmonic maps. Li et al. [22] extended this idea to the high dimensional cases, and used an iterative scheme to derive the Harmonic map which enhanced the stability of the algorithm and effectively prevented the mesh tangling. This framework has been widely used on diverse applications. A review of the method may be found in [34]. In this framework, the so-called monitor function is of particular importance since it is used to partially control the movement of the mesh grids. In the domain for the Kohn-Sham equation, compared to the distant regions from the atom, the regions around the atomic nuclei and between atoms of chemical bonds are more important [42]. Consequently, the demand of the grid points in these regions is large for resolving the wavefunctions effectively. In these important regions, the variation of the wavefunctions is much larger than that in the other regions. Therefore, the monitor function is designed based on the gradient of the wavefunctions in this paper. In practice, the smoothing of the monitor function is necessary for keeping the mesh quality. We follow [11, 40] to smooth the monitor function.

In our simulations, the Hartree potential is obtained by solving the Poisson equation with an algebraic multi-grid (AMG) method. The exchange-correlation potential is approximated by the local density approximation (LDA). The zero Dirichlet boundary condition is used for the Kohn-Sham equation, and the multipole expansion method is used to give the boundary value of the Hartree potential in the Poisson equation. To solve the generalized eigenvalue problem, the locally optimal block preconditioned conjugate gradient (LOBPCG) method [17] is used. To improve the convergence of the SCF iteration, a linear mixing scheme is adopted to update the electron density.

The outline of this paper is as follows. In the next section, the DFT and the Kohn-Sham equation are introduced. In Sect. 3, the finite element discretization of the Kohn-Sham equation is described. The mesh redistribution technique is introduced in Sect. 4. In Sect. 5, some details of numerical techniques are discussed on our algorithm. Numerical simulations are demonstrated in Sect. 6, which illustrate the reliability and the effectiveness of our method. Finally, the paper is concluded with some general remarks as well as discussions on future directions in Sect. 7.

2 The Kohn-Sham Equation

In [18], Kohn and Sham presented the following self-consistent equation to investigate the minimization of the total energy of a many-electron system (we use the atomic units $e^2 = \hbar = m = 1$ hereafter)

$$\left(\hat{T} + v_{\mathit{eff}}(\mathbf{x})\right)\psi_i(\mathbf{x}) = \varepsilon_i \psi_i(\mathbf{x}), \tag{2}$$

where $\hat{T} = -\frac{1}{2}\nabla^2$ denotes the kinetic energy, ε_i and ψ_i denote the i-th eigenvalue and wavefunction respectively, and $v_{eff}(\mathbf{x})$ is the effective potential with the following expression

$$v_{eff}(\mathbf{x}) = v_{ext}(\mathbf{x}) + v_{Hartree}(\mathbf{x}) + v_{xc}(\mathbf{x}). \tag{3}$$

The electron density can be obtained as

$$n(x) = \sum_{i}^{occ} \left|\psi(x)\right|^2, \tag{4}$$

where *occ* denotes the number of occupied states. For a closed system of M nuclei and N electrons, the external potential

$$v_{ext}(\mathbf{x}) = -\sum_{i}^{M} \frac{Z_i}{|\mathbf{x} - \mathbf{x}_i|}, \tag{5}$$

describes the attraction between the nucleus and electron. Here Z_i is the i-th nucleus charge, and $\mathbf{x}_i$ is the position of the i-th nucleus. The Hartree potential $v_{Hartree}$ is the Coulomb interaction of electrons

$$v_{Hartree}(\mathbf{x}) = \int \frac{n(\mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|} d\mathbf{x}'. \tag{6}$$

In the simulation, direct evaluating the above integral will result in $\mathcal{O}(N^2)$ operations, which is very time-consuming. A popular way to address $v_{Hartree}$ is to solve the following Poisson equation

$$-\nabla^2 v_{Hartree}(\mathbf{x}) = 4\pi n(\mathbf{x}) \tag{7}$$

with a proper Dirichlet boundary condition.

The last term in (3) is the exchange-correlation potential $v_{xc}(\mathbf{x})$. This term is caused by the Pauli exclusion principle and other non-classical Coulomb interactions. The analytical expression of the exchange-correlation potential for a general system is in general unknown. Thus this term needs to be approximated. The most popular way is to use the local density approximation (LDA) originally proposed by Kohn and Sham in [18]. The exchange-correlation energy $E_{xc}^{LDA}(n)$ is given as

$$E_{xc}^{LDA}(n) = \int \epsilon_{xc}(n) n(\mathbf{x}) d\mathbf{x}, \tag{8}$$

where $\epsilon_{xc}(n)$ stands for the exchange-correlation energy per unit volume of the homogeneous electron gas of the density n. Then v_{xc} can be addressed as

$$v_{xc}(\mathbf{x}) = \frac{\delta E_{xc}}{\delta n}. \tag{9}$$

The exchange-correlation energy E_{xc} can be separated as two parts $E_{xc} = E_x + E_c$. We adopt the expression of the exchange energy E_x proposed in [18]

$$E_x^{LDA}(n) = -\frac{3}{4}\left(\frac{3}{\pi}\right)^{1/3} \int n(\mathbf{x})^{4/3} d\mathbf{x}. \tag{10}$$

Then $v_x = -(3n(\mathbf{x})/\pi)^{1/3}$. The approximation of E_c is much more complicated than that of E_x. Even in the free electron gas, we only know the expressions of E_c of the high-density limit and the low-density limit. In the simulation, the parametrization of Perdew and Wang [30] is adopted.

The ground state energy of the physical system can be calculated from the solution of the Kohn-Sham equation:

$$E = \sum_{i}^{occ} \varepsilon_i - \int \left(\frac{1}{2} v_{Hartree} + v_{xc} \right) n(\mathbf{x}) d\mathbf{x} + E_{xc}. \tag{11}$$

Note that for the simulations of a molecule which contains several nuclei, the total energy should be added to a repulsive Coulomb term that accounts for the interactions between the nuclei

$$E_{nn} = \sum_{j,k} \frac{Z_j Z_k}{|\mathbf{x}_j - \mathbf{x}_k|}.$$

From (2), one can see that the wavefunction $\psi_i(\mathbf{x})$ depends on the effective potential $v_{eff}(\mathbf{x})$, while the electron density $n(\mathbf{x})$ defined as in (4) determines the effective potential $v_{eff}(\mathbf{x})$ in the meantime. That means Eq. (2) is non-linear, and we need an SCF method to solve it. To solve (2), many numerical methods have been proposed. Here, we focus on the finite element method for discretizing the governing equation because of its flexibility for different geometries and boundary conditions. The finite element discretization of (2) will be introduced in the next section.

3 Numerical Discretization

From now on, let us assume that the physical domain $\Omega \subset \mathcal{R}^3$ is bounded, and Ω is divided into a set of tetrahedron $\mathcal{T}$ with $\mathcal{T}_i$ as its element and $\mathbf{X}_i$ as its nodes. The piecewise linear finite element space V_h is built on $\mathcal{T}$. Note that for the linear finite element case, the interpolation points of the degrees of freedom (DOF) in each element locate on the vertexes of the tetrahedral element (3D case).

Based on the above assumptions, the wavefunction $\psi(\mathbf{x})$ in (2) can be approximated as

$$\psi_h = \sum_{i}^{N_{basis}} \psi_i \phi_i, \tag{12}$$

where N_{basis} stands for the dimension of space V_h, and ϕ_i is the i-th basis function and ψ_i is its coefficient, which is also the value of the wavefunction itself on the corresponding node.

For the finite element method, the variational approach is used to find out the coefficients $\psi = \{\psi_i\}$, $i = 1, 2, \ldots, N_{basis}$, which leads to the following linear system

$$A\psi = \varepsilon B\psi. \tag{13}$$

Here A and B are two matrices with the entries

$$A_{i,j} = \frac{1}{2} \int_{\Omega} \nabla\phi_i \cdot \nabla\phi_j d\mathbf{x} + \int_{\Omega} v_{eff} \phi_i \phi_j d\mathbf{x}, \tag{14}$$

$$B_{i,j} = \int_{\Omega} \phi_i \phi_j d\mathbf{x}, \tag{15}$$

respectively.

Equation (13) is a generalized eigenvalue problem with properties that the matrices A and B are both Hermitian, and the mass matrix B is positive definite. There are many solvers in the literature to solve this kind of problem. In our algorithm, the locally optimal block preconditioned conjugate gradient (LOBPCG) method [17] is employed.

To discretize the Poisson equation (7), we use the same finite element space mentioned above. The Hartree potential $v_{Hartree}$ can be approximated as

$$v_{H,h} = \sum_{i}^{N_{basis}} v_{H,i}\phi_i, \tag{16}$$

where $v_{H,i}$ is the coefficient of the i-th basis function ϕ_i. Let $v_H = \{v_{H,i}\}, i = 1, 2, \ldots, N_{basis}$, then the final system can be read as

$$Pv_H = f, \tag{17}$$

where P is the stiffness matrix with entry

$$P_{i,j} = \int_{\Omega} \nabla\phi_i \cdot \nabla\phi_j d\mathbf{x}, \tag{18}$$

and the right hand side f is a vector with entry

$$f_i = \int_{\Omega} 4\pi n(\mathbf{x})\phi_i d\mathbf{x}. \tag{19}$$

Note that in the above integration, the evaluation of the electron density $n(\mathbf{x})$ on a quadrature point $\mathbf{x}_q$ is given by

$$n(\mathbf{x}_q) = \sum_{i}^{occ} \sum_{j,k}^{N_{basis}} \phi_j\psi_{i,j}(\mathbf{x}_q)\phi_k\psi_{i,k}(\mathbf{x}_q).$$

An efficient solver for the Poisson equation is important, because (17) needs to be solved in each SCF iteration. In our algorithm, the algebraic multi-grid method is adopted.

The electron density is obtained by an SCF iteration. That is, the wavefunction ψ is solved from (13), and then the electron density $n(\mathbf{x})$ is obtained from (4). Then the effective potential in (2) is evaluated in terms of the new $n(\mathbf{x})$. To end up the SCF iteration, the following criterion is used

$$\left\| n_{k+1}(\mathbf{x}) - n_k(\mathbf{x}) \right\|_2 < tol, \tag{20}$$

where $n_k(\mathbf{x})$ and $n_{k+1}(\mathbf{x})$ mean the electron densities obtained from two adjacent iterations, and tol is an user-defined tolerance. Because there are a lot of ingredients which may affect the convergence of an SCF iteration such as the quality of the initial guess of the wavefunction, the complexity of the electronic structure, and the mixing scheme used in the iteration, it is nontrivial to get a convergent result. We use $tol = 1.0e-04$ in numerical simulations.

Once the self-consistent electron density is obtained, an adaptive technique will be employed to optimize the distribution of the mesh grids. Then the SCF iteration will be implemented on this new mesh, and more accurate numerical results can be expected. In the next section, the mechanism of this adaptive technique is introduced.

4 Mesh Redistribution

The mesh redistribution method in this paper is based on the harmonic maps, which can generate the nonuniform mesh from a uniform one. In [5], a framework for the movement of the mesh grids is proposed based on the harmonic mapping. Then it is effectively extended to the high dimensional cases in [22] and [23]. In the following subsection, the harmonic map is briefly introduced first. Then detailed implementation of this method is presented.

4.1 Harmonic Maps

Suppose there are two compact Riemannian manifolds Ω and Ω_c with the metric tensors d_{ij} and $r_{\alpha\beta}$ in certain local coordinates $\mathbf{x}$ and $\boldsymbol{\xi}$. Define a map $\boldsymbol{\xi} = \boldsymbol{\xi}(\mathbf{x})$ between Ω and Ω_c, then the energy of this map is given by

$$E(\boldsymbol{\xi}) = \frac{1}{2}\int_{\Omega} \sqrt{d} d^{ij} r_{\alpha\beta} \frac{\partial \xi^{\alpha}}{\partial x^i} \frac{\partial \xi^{\beta}}{\partial x^j} d\mathbf{x}, \tag{21}$$

where $d = \det(d_{ij})$, $(d_{ij}) = (d^{ij})^{-1}$, and the standard summation convention is assumed. If the map $\boldsymbol{\xi}$ is an extremum of (21), $\boldsymbol{\xi}$ is called a harmonic map. This map is the solution of the following Euler-Lagrange equation

$$\frac{1}{\sqrt{d}} \frac{\partial}{\partial x^i} \sqrt{d} d^{ij} \frac{\partial \xi^k}{\partial x^j} + d^{ij} \Gamma^k_{\beta\gamma} \frac{\partial \xi^{\beta}}{\partial x^i} \frac{\partial \xi^{\gamma}}{\partial x^j} = 0,$$

where

$$\Gamma^k_{\beta\gamma} = \frac{1}{2} r^{k\lambda} \left[\frac{\partial r_{\lambda\beta}}{\partial \xi^{\gamma}} + \frac{\partial r_{\lambda\gamma}}{\partial \xi^{\beta}} - \frac{\partial r_{\beta\gamma}}{\partial \xi^{\lambda}} \right]$$

is the Christoffel symbol of the second kind. Theoretically, the existence and uniqueness of the harmonic map $\boldsymbol{\xi}$ is guaranteed by the Hamilton-Schoen-Yau theorem [9, 31] when Ω_c is nonpositive and its boundary $\partial\Omega_c$ is convex. These requirements are easily satisfied in practice. First of all, both Ω and Ω_c are defined in the three dimensional Euclidean space, the curvature of Ω_c is zero which is obviously nonpositive. A simple choice of Ω_c in the numerical simulations is a unit cube, which is obviously convex for its boundary. Furthermore, the employment of the Euclidean space causes $\Gamma^k_{\beta\gamma} = 0$. Hence the above Euler-Lagrange equation becomes

$$\frac{\partial}{\partial x^i} G^{ij} \frac{\partial \xi^k}{\partial x^j} = 0, \tag{22}$$

and corresponding energy becomes

$$E(\boldsymbol{\xi}) = \sum_k \int_{\Omega} G^{ij} \frac{\partial \xi^k}{\partial x^i} \frac{\partial \xi^k}{\partial x^j} d\mathbf{x}, \tag{23}$$

where $G^{ij} = \sqrt{d} d^{ij}$. Following [23], we call Ω the physical domain, Ω_c the logical domain, and $M = (G^{ij})^{-1}$ the monitor function.

4.2 Mesh Redistribution Using Harmonic Maps

Suppose the SCF iteration has been finished on current mesh, and the self-consistent electron density and wavefunctions have been obtained. What we do in the next step is to reasonably redistribute the mesh grids in terms of those self-consistent quantities, with the help of the harmonic map.

In practice, the harmonic map is reached by an iterative procedure. Assume that an initial (fixed) mesh on the logical domain Ω_c is given, the general framework of generation of the harmonic map can be described as the following.

Flow Chart of the Algorithm 1

S.1 obtain the difference between the current mesh which is obtained from solving (24) and the initial mesh in the logical domain Ω_c. If the L^∞ norm of the difference is smaller than a given tolerance, the mesh redistribution is done, otherwise, go to next step;

S.2 obtain the directions and the magnitudes of the movement of mesh grids in the physical domain Ω in terms of the difference given in **S.1**, and then update the mesh in the physical domain;

S.3 update solutions on the updated mesh in the physical domain, then go to **S.1**.

In [22], the system which is used to generate the mesh in the logical domain in **S.1** is (22) together with a Dirichlet boundary condition. However, with this system, only the interior grids are redistributed, while the positions of the grid points on the boundary are unchanged. Under this situation, the redistribution of grids on the boundary of the physical domain should be handled separately, such as using homogeneous Neumann boundary conditions, extrapolating the interior mesh points to the boundary, and relocating the mesh points by solving a lower-dimensional moving-mesh PDE. However, all these strategies may affect the efficiency of the algorithm.

To redistribute the mesh points in a uniform manner, Li et al. [23] proposed an optimization problem for the harmonic map in place of the boundary value problem in [22]. The optimization problem can be read as

$$\begin{cases} \min \sum_k \int_\Omega G^{ij} \dfrac{\partial \xi^k}{\partial x^i} \dfrac{\partial \xi^k}{\partial x^j} d\mathbf{x}, \\ \text{s.t. } \xi|_{\partial\Omega} = \xi_b \in \mathbf{K}, \end{cases} \tag{24}$$

where $\mathbf{K} = \{\xi_b \in C^0(\partial\Omega) | \xi_b : \partial\Omega \to \Omega_c; \xi_{b|A_i}$ is a piecewisely linear mapping without degeneration of the Jacobian$\}$ denotes a mapping set from $\partial\Omega$ to $\partial\Omega_c$. By solving the above optimization problem with different G^{ij}, the interior mesh points and ones on the boundary move simultaneously in the logical domain Ω_c.

In the following, details in each step of Algorithm 1 will be briefly demonstrated. Assume that the tetrahedral mesh in the physical domain Ω is denoted by $\mathcal{T}$, with $\mathcal{T}_i$ as its element, and X_i as its node. For the logical domain Ω_c, we use $\mathcal{T}_c$ to denote the corresponding tetrahedral mesh, and $\mathcal{T}_{i,c}$ and $\mathcal{A}_i$ as its element and node, respectively.

To obtain the initial mesh in the logical domain Ω_c, we need to generate the initial mesh $\mathcal{T}_c^{ini}$ in the logical domain Ω_c by solving the following optimization problem

$$\begin{cases} \min \sum_k \int_\Omega \sum_i (\dfrac{\partial \xi^k}{\partial x^i})^2 d\mathbf{x}, \\ \text{s.t. } \xi|_{\partial\Omega} = \xi_b \in \mathbf{K}. \end{cases} \tag{25}$$

This initial mesh $\mathcal{T}_c^{ini}$ is only used as a reference, and it is unchanged during the whole procedure.

In the first step of the Algorithm 1, the optimization problem (24) needs to be solved. Note that at the current stage, we only try to improve the mesh quality i.e., not the physical solution, the efficiency of the solver for (24) is more important than its accuracy. However, the linear system which is obtained from (24) is neither Hermitian nor positive definite,

which takes difficulty for the efficient solver. To resolve this difficulty, it was suggested in [23] that the system be decoupled into two smaller systems: one is for the grid points on the boundary, and the other one is for the interior grid points. Although there is also no good property for the first system, it is much smaller than the original system, and a generalized minimal residual method (GMRES) or a biconjugate gradient (BiCG) method can be used to solve it. For the second one, a multi-grid method is adopted to solve it because it has good properties (Hermitian and positive definite).

To further improve the efficiency of the solver for (24), Di et al. [4] proposed a new approach to speed up the implementation. This new approach is based on an algebraic multi-grid method, together with a constraint for grid points on the boundary. In our simulations, this technique is adopted for solving (24). We refer to [4] and references therein for details.

After (24) is solved, we obtain a mesh $\mathcal{T}_c$ in the logical domain Ω_c. If the L^∞ norm of the difference between $\mathcal{T}_c$ and $\mathcal{T}_c^{ini}$ is small enough, say,

$$\left\|\mathcal{T}_c - \mathcal{T}_c^{ini}\right\|_L^\infty < tol,$$

the iteration is stopped, otherwise we will use the difference between $\mathcal{T}_c$ and $\mathcal{T}_c^{ini}$ to generate the movement of mesh grids in physical domain Ω.

In Step 2 of Algorithm 1, we use the following formula to generate the direction and magnitude of movement for each mesh grid in the physical domain,

$$\delta X_i = \frac{\sum_{\mathcal{T}} |\mathcal{T}| \frac{\partial \mathbf{x}}{\partial \xi}|_{\text{in}\mathcal{T}} \delta A_i}{\sum_{\mathcal{T}} |\mathcal{T}|},$$

where $\delta \mathcal{A}_i = \mathcal{A}_i^{ini} - \mathcal{A}_i$, and $\mathcal{T}$ stands for the element in the physical domain which has X_i as one of the vertex, and $|\mathcal{T}|$ is its volume. In the above formula, $\partial \mathbf{x}/\partial \xi$ is given by solving the following system in each element,

$$\begin{pmatrix} A^1_{\mathcal{T}_c,1} - A^1_{\mathcal{T}_c,0} & A^1_{\mathcal{T}_c,2} - A^1_{\mathcal{T}_c,0} & A^1_{\mathcal{T}_c,3} - A^1_{\mathcal{T}_c,0} \\ A^2_{\mathcal{T}_c,1} - A^2_{\mathcal{T}_c,0} & A^2_{\mathcal{T}_c,2} - A^2_{\mathcal{T}_c,0} & A^2_{\mathcal{T}_c,3} - A^2_{\mathcal{T}_c,0} \\ A^3_{\mathcal{T}_c,1} - A^3_{\mathcal{T}_c,0} & A^3_{\mathcal{T}_c,2} - A^3_{\mathcal{T}_c,0} & A^3_{\mathcal{T}_c,3} - A^3_{\mathcal{T}_c,0} \end{pmatrix} \begin{pmatrix} \frac{\partial x^1}{\partial \xi^1} & \frac{\partial x^1}{\partial \xi^2} & \frac{\partial x^1}{\partial \xi^3} \\ \frac{\partial x^2}{\partial \xi^1} & \frac{\partial x^2}{\partial \xi^2} & \frac{\partial x^2}{\partial \xi^3} \\ \frac{\partial x^3}{\partial \xi^1} & \frac{\partial x^3}{\partial \xi^2} & \frac{\partial x^3}{\partial \xi^3} \end{pmatrix} = \begin{pmatrix} X^1_{\mathcal{T},1} - X^1_{\mathcal{T},0} & X^1_{\mathcal{T},2} - X^1_{\mathcal{T},0} & X^1_{\mathcal{T},3} - X^1_{\mathcal{T},0} \\ X^2_{\mathcal{T},1} - X^2_{\mathcal{T},0} & X^2_{\mathcal{T},2} - X^2_{\mathcal{T},0} & X^2_{\mathcal{T},3} - X^2_{\mathcal{T},0} \\ X^3_{\mathcal{T},1} - X^3_{\mathcal{T},0} & X^3_{\mathcal{T},2} - X^3_{\mathcal{T},0} & X^3_{\mathcal{T},3} - X^3_{\mathcal{T},0} \end{pmatrix},$$

where $(X^1_{\mathcal{T},i}, X^2_{\mathcal{T},i}, X^3_{\mathcal{T},i})$ means the i-th vertex of the element $\mathcal{T}$ in physical domain Ω, and $(A^1_{\mathcal{T}_c,i}, A^2_{\mathcal{T}_c,i}, A^3_{\mathcal{T}_c,i})$ means the i-th vertex of its corresponding element $\mathcal{T}_c$ in logical domain Ω_c. Because the topological structure of the mesh will not change during the iteration, the correspondence between the meshes in Ω and Ω_c will also not change.

After we get δX_i, the grid point X_i in the physical domain Ω is updated by

$$X_i^{new} = X_i + \tau \delta X_i,$$

where $\tau \in [0, 1]$. Here τ is used to prevent the mesh tangling.

The update of the wavefunctions is designed under the assumption that the surface of the numerical solution does not move during the mesh redistribution. For the details in the implementation, we refer to [22]. It worth mentioning that this method is one kind of interpolation-free solution-updating methods, which makes the implementation much simpler than the interpolation methods, especially for the tetrahedral mesh case. In fact, only

an ordinary differential equation is solved for each wavefunction in the update process. Furthermore, the requirement on the numerical accuracy is not crucial for our time-independent problem because a rough solution update still give us a quite good approximation for the next SCF iteration. Consequently, the low-order, highly efficient numerical schemes such as the Euler schemes, can be adopted to solve that ODE system. A potential issue for this update method is that the electron density may no longer be conserved in the physical domain. To avoid this issue, each wavefunction is normalized before the SCF iteration.

The monitor function in (24) is important in the simulations. A good monitor function can help on resolving the important regions in the physical domain with the mesh grids, while keeps the mesh quality. In the following subsection, a monitor function is given, and a special smoothness strategy of the monitor function is also discussed.

4.3 Monitor Function

As mentioned in the introduction, the important regions in the simulation of the Kohn-Sham equation are the vicinity of nucleus and between atoms of chemical bonds. Based on our numerical experience, the variation of the wavefunctions in those regions is much larger than that in other regions. Therefore, we choose the following monitor function

$$M = \left(\sqrt{\epsilon + \sum_{i}^{occ} |\nabla \psi_i(\mathbf{x})|^2} \right) I, \tag{26}$$

where I stands for an identity matrix. With (26), the grid points move towards the region with large gradient of the wavefunctions, and the adaptivity is controlled by the parameter ϵ. The smaller ϵ results in more adaptivity. Note that the problem is non-periodic, a sufficiently large domain is needed to reduce the truncation error. At the meantime, the large variation of the density only appears in the vicinity of the nucleus, which is a relatively small region. Consequently, the monitor function (26) results in a dramatic change of the mesh size around the nucleus. Figure 1(a) and (b) demonstrates this phenomena. Two drawbacks are observed from the figure: (i), although the density of grid points in the vicinity of the nucleus becomes large after the movement of the mesh, the large distortion of the element is also observed near the nucleus, which may negatively affects on the quality of the numerical results and the convergence of the SCF iteration; (ii), there are too many grid points in the outer layer of the domain. Because there is almost no variation of the wavefunctions far away from the nucleus, the positions of grid points in that region are almost unchanged by using the monitor function (26). Based on the above two observations, moving the grid points in the region away from the nucleus towards to the important regions is a good idea to further improve the mesh quality.

A similar situation which is mentioned above has been discussed by Wang et al. in [40]. They avoid this problem by introducing a diffusive process in the monitor function. This strategy is also adopted by Hu and Zegeling in [11]. In the following, we will give the implementation of this technique in our algorithm.

Besides those two terms in (26), a new term $\tilde{n}$ which is the solution of the following equation

$$\frac{\partial \tilde{n}}{\partial t} - \delta \Delta \tilde{n} = 0 \tag{27}$$

is added, where δ is a parameter which depends on the size of the physical domain. The initial value of $\tilde{n}$ is given by $\tilde{n}^0 = \sum_i^{occ} |\nabla \psi_i(\mathbf{x})|^2$. With the function $\tilde{n}$, the monitor function

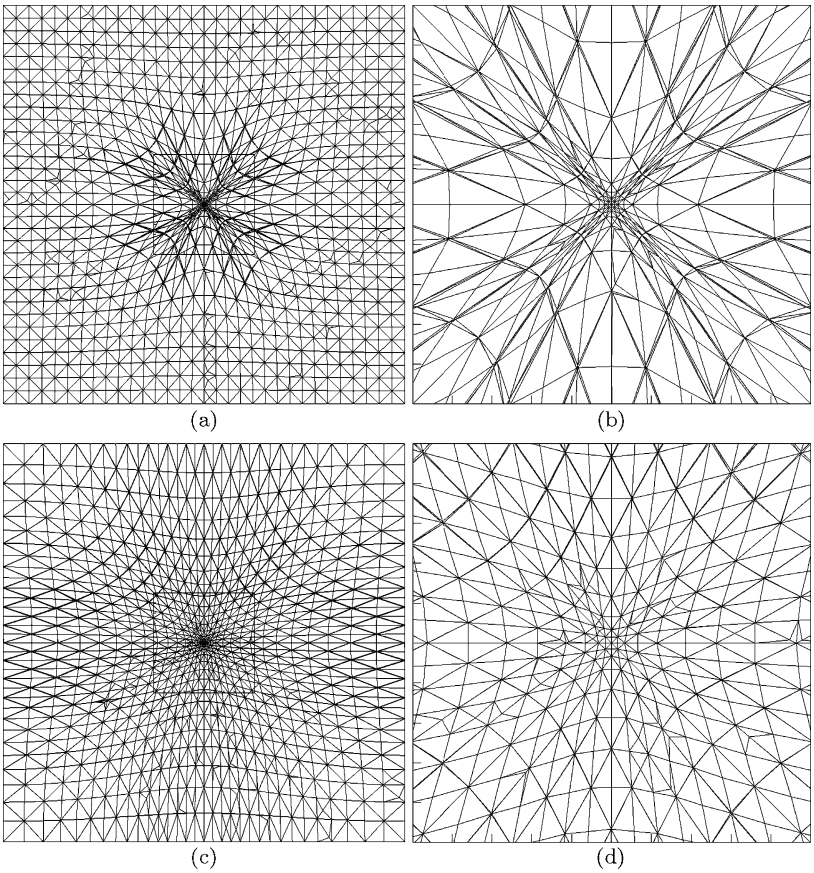

Fig. 1 The demonstration of a slice of the 3-dimensional mesh in the plane $(0, 0, 1)$ after the mesh redistribution. (**a**) and (**b**) are generated with (26), while (**c**) and (**d**) are generated with (28). (**b**) and (**d**) show the detailed mesh around the nucleus of the whole mesh (**a**) and (**c**) respectively

becomes the following form

$$M = \left(\sqrt{\epsilon + \tilde{n} + \sum_{i}^{occ} \left| \nabla \psi_i(\mathbf{x}) \right|^2} \right) I. \tag{28}$$

Because we only want a rough solution for (27), the variable $\tilde{n}$ is just propagated one or two times, the increment of the CPU time which is caused by solving (27) is not significant, with the help of the AMG method. The significant improvement of the mesh can be observed from Fig. 1(c) and (d).

We close this section with the following remarks.

– Besides the harmonic mapping method, there are also other methods which are used to realize the mesh redistribution. We refer to [13] and references therein for those methods.
– Note that the monitor function M in (28) is defined by a scalar value multiplying an identity matrix. This results in an isotropic mesh in our simulations. In fact, the anisotropic mesh can also be generated by using this framework with an anisotropic monitor function [13].

5 Computational Issues

We have stated the finite element discretization of the Kohn-Sham equation, and introduced Algorithm 1 to improve the quality of the numerical results. In this section, we present a few of the technical details in the algorithm, and the flow chart of the whole algorithm at the end.

5.1 Boundary Conditions

To solve the equations (13) and (17), appropriate boundary conditions are needed. For (13), the zero Dirichlet boundary conditions are adopted naturally when the size of the physical domain Ω is sufficiently large. However, for (17), because the Hartree potential decays as N/r, where N is the electron number, the simple use of zero Dirichlet boundary condition will introduce large truncation error on the boundary. To give the evaluation of the Hartree potential on the boundary, (6) can be used directly. However, it results in a $\mathcal{O}(N^{\frac{5}{3}})$ operations in the algorithm, which is not consistent with $\mathcal{O}(N)$ of the AMG solver. To reduce the cost, a multipole expansion approximation is employed for the boundary values. In the simulations, the following approximation is used

$$V_{Hartree}(\mathbf{x})|_{\mathbf{x}\in\partial\Omega} \approx \frac{1}{|\mathbf{x}-\mathbf{x}''|}\int_{\Omega} n(\mathbf{x}')d\mathbf{x}' + \sum_{i=1,2,3} p_i \cdot \frac{x^i - x''^{,i}}{|\mathbf{x}-\mathbf{x}''|^3} + \sum_{i,j=1,2,3} q_{ij} \cdot \frac{3(x^i - x''^{,i})(x^j - x''^{,j}) - \delta_{ij}|\mathbf{x}-\mathbf{x}''|^2}{|\mathbf{x}-\mathbf{x}''|^5}, \tag{29}$$

where

$$p_i = \int_{\Omega} n(\mathbf{x}')(x'^{,i} - x''^{,i})d\mathbf{x}', \qquad q_{ij} = \int_{\Omega} \frac{1}{2} n(\mathbf{x}')(x'^{,i} - x''^{,i})(x'^{,j} - x''^{,j})d\mathbf{x}'.$$

In the above expressions, $\mathbf{x}''$ stands for an arbitrary point in Ω. In the implementation, the following choice

$$\mathbf{x}'' = \frac{\int \mathbf{x} n(\mathbf{x})d\mathbf{x}}{\int n(\mathbf{x})d\mathbf{x}}$$

is always a reasonable one.

Note that the accuracy of the approximate Hartree potential at a point $\mathbf{x}$ by using the multipole expansion depends on the distance between $\mathbf{x}$ and $\mathbf{x}''$, which means that a good approximation can be expected when $|\mathbf{x}-\mathbf{x}''|$ is sufficiently large. Consequently, a sufficiently large computational domain is needed in the numerical simulations. To investigate the relation between the numerical error and the size of the domain when the multipole expansion method is adopted, the following numerical experiment is presented. Suppose the electron distribution is given as

$$n(\mathbf{x}) = \sum_{i=1,2} Z_i \left(\frac{b_i}{\pi}\right)^{3/2} e^{-b_i|\mathbf{x}-\mathbf{a}_i|^2},$$

which imitates the electron distribution of a diatomic molecule. Here $\mathbf{a}_1 = (2,0,0)$, $\mathbf{a}_2 = (-2,0,0)$, $b_1 = 0.8$, $b_2 = 0.6$, $Z_1 = 6$, and $Z_2 = 4$. For this electron distribution, the analytical Hartree potential is given by

$$v_{Hartree} = \sum_{i=1,2} Z_i \frac{erf(\sqrt{b_i}|\mathbf{x}-\mathbf{a}_i|)}{|\mathbf{x}-\mathbf{a}_i|},$$

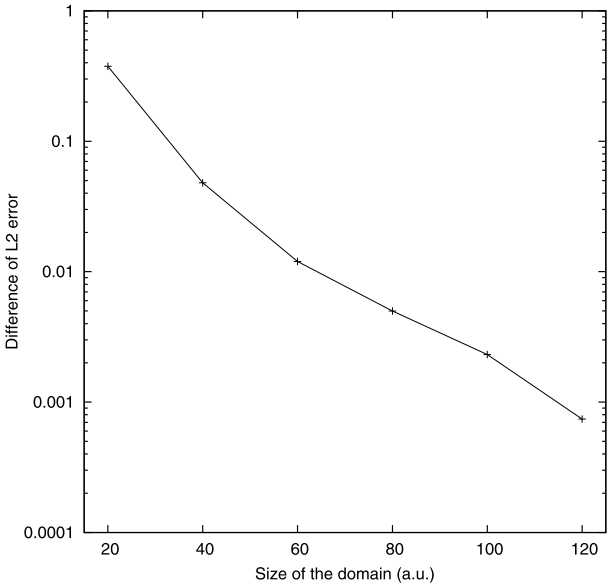

Fig. 2 The relation between the L_2 error of the approximate Hartree potential and the domain size

where *erf* stands for the error function. To numerically solve the above problem, the finite element method which is described in Sect. 3 is employed. The exact boundary value and the approximate one which is given by (29) are considered respectively. The computational domain is a cube, and different sizes are used, from $[-10, 10]^3$ to $[-60, 60]^3$. Here, the atomic unit (a.u.) is used. For the mesh, the sizes of the tetrahedral element around the origin point are approximately kept, only the size of the domain is enlarged. An algebraic multigrid solver is developed to efficiently solve the resulting linear system. This is also the solver we used in the algorithm for generating the Hartree potential. The difference of the L_2 error between the exact one and the approximate one is demonstrated in Fig. 2.

From Fig. 2, it can be obviously observed that the difference of the L_2 error between the exact one and the approximate one becomes small when the size of the computational domain becomes large. It is only $7.0e-04$ when the size $[-60, 60]^3$ is adopted, which shows that the multipole expansion method works very well in the implementation. However, to balance the accuracy of the numerical solutions and the efficiency of the algorithm, the selection of the size of the computational domain is problem dependent in the simulations.

5.2 Self-Consistent Field Iteration

It is nontrivial to get the convergence for the SCF iterative procedure. For the convergence analysis of the SCF iteration, we refer to [41] and references therein.

The commonly used techniques for helping the SCF convergence are the extrapolation method, the damping method, the level shifting method, and the direct inversion in the iterative subspace (DIIS) method, etc. Among these techniques, the DIIS method has proved to be very efficient in forcing the convergence and in reducing the number of iterations at the same time. However, it demands considerable memory to save a sequence of wavefunctions in previous iteration steps, especially when a complicated electronic structure is simulated. Comparatively speaking, the damping method is quite efficient because only the wavefunctions in the last iteration step are needed. The damping method can effectively prevent the divergence which is caused by the oscillations. In our simulations, the following damping

method is adopted. Suppose that in the k-th iteration step, the new electron density $\tilde{n}_{k+1}$ has been calculated from the electron density n_k. Then the following formula is used to generate n_{k+1}.

$$n_{k+1} = \gamma \tilde{n}_{k+1} + (1-\gamma) n_k, \tag{30}$$

where $0 < \gamma < 1$. In the simulations, we always use $\gamma = 0.7$. In all of our numerical simulations, the convergence of the SCF iteration is successfully achieved with the help of the above damping method. Note that we let $tol = 1.0e-04$ in (20).

Now we are ready to give the flow chart of the whole algorithm.

Flow Chart of the Algorithm 2

Input: The initial guess ψ_i^{ini} for the wavefunctions. Let $n_0(\mathbf{x}) = n^{ini}(\mathbf{x})$, and $i, k = 0$.

S.1 With $n_k(\mathbf{x})$, get the evaluation of $v_{Hartree}$ by solving (17), and the evaluations of v_{ext} and v_{xc} with (5) and (9) respectively. Then the generalized eigenvalue problem (13) is obtained.

S.2 Using LOBPCG method to solve (13) to obtain the ψ_{k+1} and $\tilde{n}_{k+1}(\mathbf{x})$. If (20) is not satisfied, update $n_{k+1}(\mathbf{x})$ by using (30), and let $k = k+1$, then goto **S.1**; Otherwise, calculate the approximate ground state energy E_i by using (11), and goto **S.3**

S.3 If $|E_i - E_{i-1}| < etol$, goto *Output*;
Otherwise, use Algorithm 1 to redistribute the grid points and update the solutions, and goto **S.1**.

Output: the approximation of the ground state density, the total energy, and wavefunctions.

In the above algorithm, E_i and E_{i+1} are the energies of the ground state from two adjacent iteration steps, and in the implementation, $etol = 1.0e-02$ is used.

6 Numerical Examples

In this section, the convergence of the proposed method is examined first. Then several simulations are implemented to show the reliability and effectiveness of our method. All simulations are implemented by using a C++ library AFEABIC (Adaptive Finite Element package for the AB-Initio Calculations). In this library, the Libxc [26] and APE (Atomic Pseudopotential Engine) [28] are employed for the generation of the exchange-correlation potential and the pseudopotential, respectively. Currently, this library is still under development by the authors. The hardware configuration is a personal desktop with Intel Pentium 3.33 GHz CPU and 4 GB of RAM. For all simulations in this section, the wavefunctions are initialized randomly.

Example 1 Solve the equation

$$-\frac{1}{2}\nabla^2 u - \frac{1}{|\mathbf{x}|}u = \lambda u \tag{31}$$

with Algorithm 2. The lowest eigenvalue of this equation is -0.5.

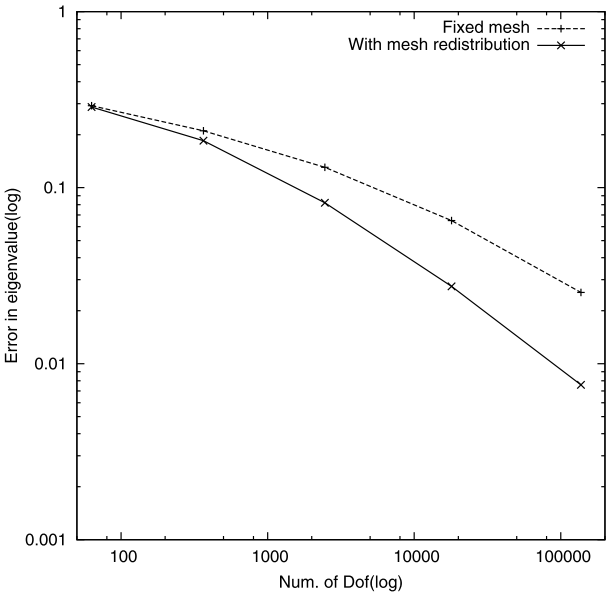

Fig. 3 Convergence rate of the eigenvalue with fixed uniform mesh and with mesh redistribution for Example 1

The above equation models the hydrogen atom. The physical domain is chosen as $[-10, 10] \times [-10, 10] \times [-10, 10]$. The uniform mesh is refined successively from the coarsest one which contains 63 grid points. The results are demonstrated in Fig. 3, which shows that both the convergence rate and the accuracy obtained with the mesh redistribution are superior to that of the fixed uniform mesh. With the refinement of the mesh, the convergence order of solver with mesh redistribution reaches around 1.9, which shows that the eigenvalue of this equation converges at the expected rate of the convergence for the linear finite element method.

Example 2 Simulate the selected atoms by using the all-electron calculation.

First of all, the helium atom is studied in detail. For the all-electron calculation, the external potential (5) is very singular around the nucleus. An initial uniform mesh may result in a large number of mesh grids to effectively resolve this singularity, which cause difficulties for convergence study of our method. Therefore, a rough radial mesh is used in the simulation. The coarsest mesh has 505 nodes, and the mesh is successively refined three times to investigate the convergence of the total energy of the ground state. Besides the numerical accuracy, the CPU time needed in the simulation is also studied. The results are shown in Fig. 4, and the following two observations can be made from the results.

- With the successive refinement of the mesh, both results (with and without mesh redistribution) converge to the reference energy, -2.83 a.u. for helium atom [19]. Furthermore, with the same number of the mesh grids, the mesh redistribution method significantly improves the accuracy of the numerical solution.
- To get the same accuracy of the numerical solution, our mesh redistribution method always needs less mesh grids and less CPU time, compared with fixed mesh case. For example, much more than 26903 mesh grids and 115 CPU seconds are needed by fixed mesh case to get the total energy -2.63 a.u., while it can be done only with around 3568 mesh grids and 32 CPU seconds with the help of our mesh redistribution method. In other words, both the storage and the CPU time required by the simulation are effectively saved simultaneously, with our mesh redistribution method.

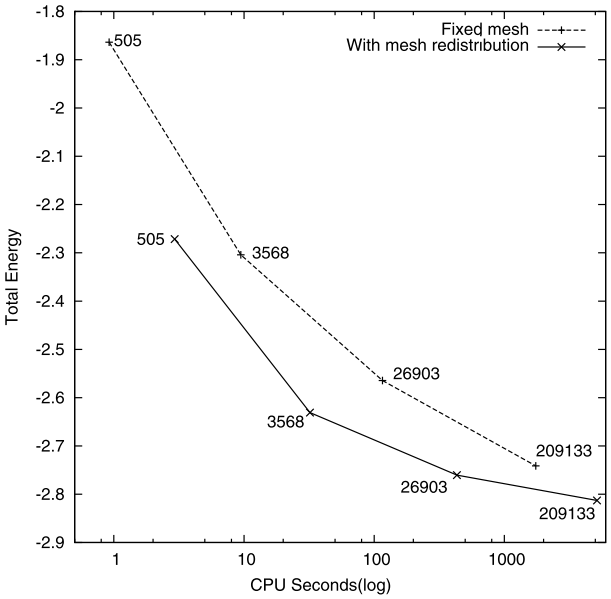

Fig. 4 Numerical results for helium atom in Example 2. The number in the figure stands for the number of mesh grids in each simulation. The reference value for the total energy is −2.83 a.u.

Table 1 Ground state energies of selected atoms (a.u.)

	This work	KS-LSDA [19]	Experiment [39]
Helium	−2.83	−2.83	−2.904
Lithium	−7.33	−7.34	−7.478
Boron	−24.30	−24.34	−24.6579

There are two methods which can further reduce the CPU time. The first one is to use a better initial guess for the wavefunction. A random initialization for the wavefunctions is adopted in the simulation, so the initial SCF iteration is needed for generating the reliable monitor function. If the wavefunction is well-initialized, this first SCF iteration can be avoided. Also, a better initialization can effectively accelerate the convergence of the SCF iteration. The second one is to use a big tolerance in the first SCF iteration. With a big tolerance, the number of the SCF iteration can be effectively reduced. After the first SCF iteration, a small tolerance is adopted for the numerical accuracy. With the above two methods, the CPU time can be further reduced.

Finally, the total ground state energies of some selected atoms are shown in Table 1. It can be obviously seen that our algorithm gives very good approximation.

Example 3 Simulate the selected atoms by using the Troullier-Martin [36] pseudopotential.

In the simulations, the norm-conserving Troullier-Martin pseudopotential is employed. To enhance the efficiency of the implementation of the pseudopotential calculations, the Kleinman-Bylander [16] form is used. The pseudoatom energies of some selected atoms are calculated, and the results are given in Table 2. As a comparison, the corresponding experimental values are also given in the table. It is observed that our numerical results agree with the experimental ones very well.

Table 2 Pseudo-atom energies of selected atoms (eV)

	This work	Experiment [7]
Lithium	−5.32	−5.33
Beryllium	−27.22	−27.53
Sodium	−5.13	−5.14
Magnesium	−22.78	−22.68
Aluminum	−53.17	−53.26

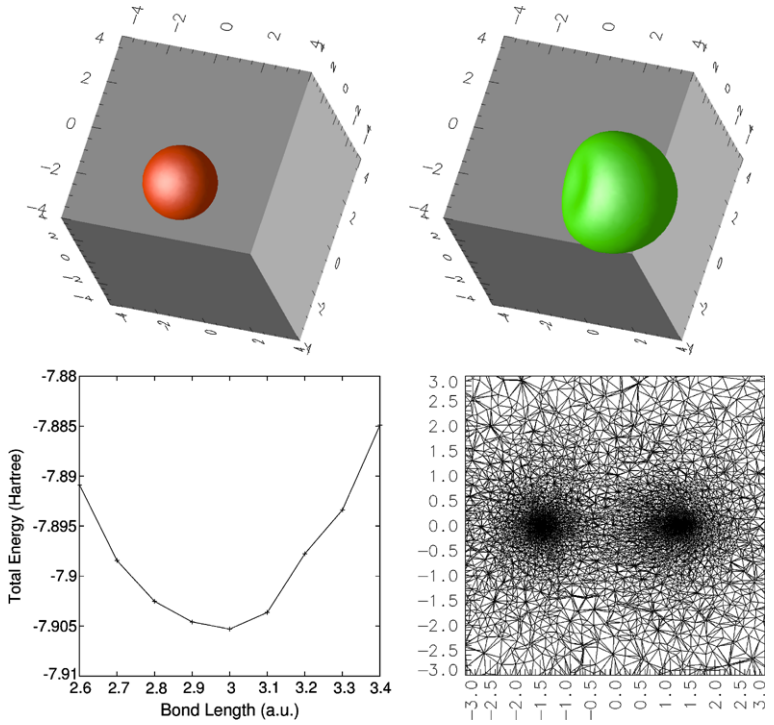

Fig. 5 *Top left*: The 1*s* orbital of the lithium atom; *Top right*: σ (sigma) bonding orbital of the lithium hydride which consists of the 1*s* orbital of the hydrogen atom and the 2*s* orbital of the lithium atom; *Bottom left*: The relation between the total energy of the lithium hydride and the bond length of Li–H; *Bottom right*: The mesh with the mesh redistribution (The sliced mesh on the X-Y plane)

Example 4 The all-electron simulation of the molecule lithium hydride (LiH).

From the structure of the molecule LiH, the variation of the electron density is large around the lithium atom and the hydrogen atom, and in the region between the lithium atom and the hydrogen atom. The result in Fig. 5 (bottom right) shows that our mesh redistribution technique adaptively resolves these important regions successfully.

From the molecular orbital theory, there are two orbitals in the LiH. The first one is the 1*s* orbital of the lithium atom. The second one is σ bonding orbital, and it consists of the 1*s* orbital of the hydrogen atom and the 2*s* orbital of the lithium orbital. Because there is no more orbital for the σ antibonding orbital, the molecule, lithium hydride, is stable. Figure 5 (top) shows the above two orbitals, respectively.

The relationship between the bond length of the Li–H and the total energy of the ground state of the molecule LiH is also studied by using our solver. The result is shown in Fig. 5 (bottom left). It can be observed easily from the curve that the bond length of the Li–H is very close to 3.0 a.u. This agrees with the experimental value 3.0107 a.u. [15] very well.

7 Concluding Remarks

We present a finite element method with an adaptive mesh redistribution technique to solve the Kohn-Sham equation in this paper. The solver includes two main iterations. The first one is an SCF iteration which is for the calculations of the self-consistent electron density on the current mesh. In this iteration, the Kohn-Sham equation is discretized by the standard finite element method. In the Hamiltonian operator, the Hartree potential is obtained by solving the Poisson equation with AMG method. The LDA is used to give the exchange-correlation potential. For the external potential, both the all-electron and the local pseudopotential are considered. The generalized eigenvalue problem is solved using the LOBPCG method.

After the self-consistent electron density is obtained, an adaptive mesh redistribution technique, which is based on the harmonic mapping, is proposed to optimize the mesh quality. The harmonic mapping is obtained iteratively with a given monitor function, which depends on the gradient of the electron density. To guarantee the mesh quality, the monitor function is smoothed by a method which is based on the diffusive mechanism. The results show that the mesh quality is significantly improved with our mesh redistribution strategy.

The numerical experiments successfully demonstrate the convergence of our solver. Furthermore, the numerical accuracy and the CPU time are also studied. Results show that with the help of the mesh redistribution technique, both the numerical accuracy of the solution and the efficiency of the algorithm are significantly improved, compared with the fixed mesh case.

In this paper, we mainly focus on the ground state of atoms and molecules. To obtain the accurate ground state energy and the electron density, using a well-designed non-uniform mesh is proven a good idea by [21, 32]. It is expected that our solver may be extended to time-dependent density functional theory (TDDFT) case. In the TDDFT, if the external potential is not strong, the system can be studied by the perturbative methods. Under this situation, the perturbed system is not far away from the ground state, which means that the well-designed, fixed mesh is still applicable. However, this may not be the case for a very strong external potential. For example, for the simulations of the high-order harmonic generations and the multi-photon ionizations. In these cases, the electronic structure may be dramatically changed. This may cause difficulty for just using a fixed mesh. Consequently, the mesh redistribution technique which is proposed in this paper may help on this issue. With our adaptive technique, the region with the large gradient of the electron density will always be resolved adaptively, which can effectively improve the numerical accuracy of solutions. Some preliminary numerical results have shown the advantages of our solver on the TDDFT calculations. The research findings will be reported on our forthcoming paper.

Acknowledgements The authors would like to thank Prof. Chao Yang (Lawrence Berkeley National Laboratory) for his helpful comments, suggestions on this work. The research was supported in part by the NSF Focused Research Group grant DMS-0968360. The research of G. Bao was supported in part by the NSF grants DMS-0908325, CCF-0830161, EAR-0724527, DMS-0968360, DMS-1211292, the ONR grant N00014-12-1-0319, a Key Project of the Major Research Plan of NSFC (No. 91130004), and a special research grant from Zhejiang University.

References

1. Baines, M.J., Hubbard, M.E., Jimack, P.K.: Velocity-based moving mesh methods for nonlinear partial differential equations. Commun. Comput. Phys. **10**, 509–576 (2011)
2. Beckett, G., Mackenzie, J.A., Robertson, M.L.: An r-adaptive finite element method for the solution of the two-dimensional phase-field equations. Commun. Comput. Phys. **1**, 805–826 (2006)
3. Chen, H.J., Zhou, A.H.: Orbital-free density functional theory for molecular structure calculations. Numer. Math. Theor. Meth. Appl. **1**, 1–28 (2008)
4. Di, Y.N., Li, R., Tang, T.: A general moving mesh framework in 3D and its application for simulating the mixture of multi-phase flows. Commun. Comput. Phys. **3**, 582–602 (2008)
5. Dvinsky, A.S.: Adaptive grid generation from harmonic maps on Riemannian manifolds. J. Comput. Phys. **95**, 450–476 (1991)
6. Echenique, P., Alonso, J.L.: A mathematical and computational review of Hartree-Fock SCF methods in quantum chemistry. Mol. Phys. **105**(23–24), 3057–3098 (2007)
7. Emsley, J.: The Elements. Oxford University Press, London (1991)
8. Fattebert, J.-L., Buongiorno Nardelli, M.: Finite difference methods for ab initio electronic structure and quantum transport calculations of nanostructures. In: Le Bris, C. (ed.) Computational Chemistry. Handbook of Numerical Analysis, vol. 10, pp. 571–612. Elsevier, Amsterdam (2003). Special volume
9. Hamilton, R.S.: Harmonic Maps of Manifolds with Boundary, vol. 471. Springer, Berlin (1975)
10. Hohenberg, P., Kohn, W.: Inhomogeneous electron gas. Phys. Rev. **136**, B864–B871 (1964)
11. Hu, G.H., Zegeling, P.A.: Simulating finger phenomenon in porous media with a moving finite element method. J. Comput. Phys. **230**, 3249–3263 (2011)
12. Hu, G.H., Qiao, Z.H., Tang, T.: Moving finite element simulations for reaction-diffusion systems. Adv. Appl. Math. Mech. **4**, 365–381 (2012)
13. Huang, W., Russell, R.D.: Adaptive Moving Mesh Methods. Applied Mathematical Sciences. Springer, Berlin (2010)
14. Hung, L., Huang, C., Carter, E.A.: Preconditioners and electron density optimization in orbital-free density functional theory. Commun. Comput. Phys. **12**, 135–161 (2012)
15. Johnson III, R.D.: NIST computational chemistry comparison and benchmark database, and NIST standard reference database (2011). http://cccbdb.nist.gov/
16. Kleinman, L., Bylander, D.M.: Effcacious form for model pseudopotentials. Phys. Rev. Lett. **48**, 1425–1428 (1982)
17. Knyazev, A.V., Argentati, M.E., Lashuk, I., Ovtchinnikov, E.E.: Block locally optimal preconditioned eigenvalue xolvers (BLOPEX) in HYPRE and PETSC. SIAM J. Sci. Comput. **29**(5), 2224–2239 (2007)
18. Kohn, W., Sham, L.J.: Self-consistent equations including exchange and correlation effects. Phys. Rev. **140**, A1133–A1138 (1965)
19. Kotochigova, S., Levine, Z.H., Shirley, E.L., Stiles, M.D., Clark, C.W.: Local-density-functional calculations of the energy of atoms. Phys. Rev. A **55**, 191–199 (1997)
20. Kresse, G., Furthmüller, J.: Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. Phys. Rev. B, Condens. Matter **54**, 11169–11186 (1996)
21. Lehtovaara, L., Havu, V., Puska, M.: All-electron density functional theory and time-dependent density functional theory with high-order finite elements. J. Chem. Phys. **131**, 054103 (2009)
22. Li, R., Tang, T., Zhang, P.W.: Moving mesh methods in multiple dimensions based on harmonic maps. J. Comput. Phys. **170**, 562–588 (2001)
23. Li, R., Tang, T., Zhang, P.W.: A moving mesh finite element algorithm for singular problems in two and three space dimensions. J. Comput. Phys. **177**, 365–393 (2002)
24. Lin, L., Lu, J., Ying, L., Weinan, E.: Adaptive local basis set for Kohn-Sham density functional theory in a discontinuous Galerkin framework I: total energy calculation. J. Comput. Phys. **231**(4), 2140–2154 (2012)
25. Marques, M., Fiolhais, C., Nogueira, F. (eds.): A Primer in Density Functional Theory. 1st edn. Springer, Berlin (2003)
26. Marques, M.A.L., Oliveira, M.J.T., Burnus, T.: Libxc: a library of exchange and correlation functionals for density functional theory. Comput. Phys. Commun. **183**(10), 2272–2281 (2012)
27. Nightingale, M.P., Umrigar, C.J. (eds.): Quantum Monte Carlo Methods in Physics and Chemistry. NATO ASI Ser. C, vol. 525. Kluwer, Dordrecht (1999)
28. Oliveira, M.J.T., Nogueira, F.: Generating relativistic pseudo-potentials with explicit incorporation of semi-core states using APE, the atomic pseudo-potential engine. Comput. Phys. Commun. **178**, 524–534 (2007)
29. Pask, J.E., Sterne, P.A.: Finite element methods in ab initio electronic structure calculations. Model. Simul. Mater. Sci. Eng. **13**(3), R71 (2005)

30. Perdew, J.P., Wang, Y.: Accurate and simple analytical representation of the electron-gas correlation energy. Phys. Rev. B, Condens. Matter **45**, 13244–13249 (1992)
31. Schoen, R., Yau, S.-T.: On univalent harmonic maps between surfaces. Invent. Math. **44**, 265–278 (1978)
32. Suryanarayana, P., Gavini, V., Blesgen, T., Bhattacharya, K., Ortiz, M.: Non-periodic finite-element formulation of Kohn-Sham density functional theory. J. Mech. Phys. Solids **58**, 256–280 (2010)
33. Suryanarayana, P., Bhattacharya, K., Ortiz, M.: A mesh-free convex approximation scheme for Kohn-Sham density functional theory. J. Comput. Phys. **230**(13), 5226–5238 (2011)
34. Tang, T.: Moving mesh methods for computational fluid dynamics. Contemp. Math. **383** (2005)
35. Torsti, T., Eirola, T., Enkovaara, J., Hakala, T., Havu, P., Havu, V., Höynälänmaa, T., Ignatius, J., Lyly, M., Makkonen, I., Rantala, T.T., Ruokolainen, J., Ruotsalainen, K., Räsänen, E., Saarikoski, H., Puska, M.J.: Three real-space discretization techniques in electronic structure calculations. Phys. Status Solidi (b) **243**, 1016–1053 (2006)
36. Troullier, N., Martins, J.L.: Efficient pseudopotentials for plane-wave calculations. Phys. Rev. B, Condens. Matter Mater. Phys. **43**, 1993–2006 (1991)
37. Tsuchida, E., Tsukada, M.: Adaptive finite-element method for electronic-structure calculations. Phys. Rev. B, Condens. Matter Mater. Phys. **54**, 7602–7605 (1996)
38. van Dam, A., Zegeling, P.A.: A robust moving mesh finite volume method applied to 1D hyperbolic conservation laws from magnetohydrodynamics. J. Comput. Phys. **216**(2), 526–546 (2006)
39. Veillard, A., Clementi, E.: Correlation energy in atomic systems. V. Degeneracy effects for the second-row atoms. J. Chem. Phys. **49**, 2415–2421 (1968)
40. Wang, H.Y., Li, R., Tang, T.: Efficient computation of dendritic growth with r-adaptive finite element methods. J. Comput. Phys. **227**(12), 5984–6000 (2008)
41. Yang, C., Gao, W.G., Meza, J.C.: On the convergence of the self-consistent field iteration for a class of nonlinear eigenvalue problems. SIAM J. Matrix Anal. Appl. **30**, 1773–1788 (2009)
42. Zhang, D.E., Shen, L.H., Zhou, A.H., Gong, X.G.: Finite element method for solving Kohn-Sham equations based on self-adaptive tetrahedral mesh. Phys. Lett. A **372**, 5071–5076 (2008)